Sequential Algorithms for Matrix and Tensor Completion

Akshay Krishnamurthy [*1] and Aarti Singh [†2]

[1]Computer Science Department, Carnegie Mellon University
[2]Machine Learning Department, Carnegie Mellon University

April 18, 2013

Abstract

We study low rank matrix and tensor completion and propose novel algorithms with the best known sample complexity guarantees for these problems. Our algorithms are *active* in that they interact with the sampling mechanism to obtain informative measurements. They are also *sequential*, processing the columns (sub-tensors) one at a time, and can easily be implemented in a streaming setting. For matrix completion, we show that one can exactly recover an $n \times n$ matrix of rank r using $O(r^2 n \log(r))$ observations and for tensor completion, one can recover an order-T tensor using $O(r^{2(T-1)} T^2 n \log(r))$ observations. We also establish a necessary condition for exact tensor completion from random observations. We complement our study with simulations that verify our theoretical guarantees and demonstrate the scalability of our algorithms.

1 Introduction

Computational resources and sensing technologies have failed to keep up with the rapid increases in the size and complexity of modern scientific investigation. Insufficient processing power restricts analysts to using heuristics and approximations that often fail to harness all of the information available in a dataset. At the same time, costs associated with data acquisition often imply that datasets are highly unreliable, either due to missing observations or low signal-to-noise ratios. Combined, both factors imply that modern data sets have low information content, and what information is available cannot even be harnessed effectively.

As a concrete example, in biological sciences, a common practice is to use an initial high throughput assay to inform subsequent investigation, the latter often involving more robust – and more expensive – data collection. While high throughput technologies offer a low-cost solution to data acquition, these data sets are often incredibly high dimensional and very noisy. Meanwhile, acquisition costs prevent biologists from using low-throughput technologies at scale, reflecting the tradeoff between reliable but costly data one one hand, and noisy but inexpensive data on the other.

In the machine learning literature, the two main solutions approach the data acquisition problem from opposite sides. On one hand, statistical techniques for many high-dimensional problems are now near-optimal in an information theoretic sense, allowing scientists to effectively leverage high throughput technologies. On the other hand, a number of researchers have shown that we can mitigate data acquisition

[*]akshaykr@cs.cmu.edu
[†]aarti@cs.cmu.edu

costs, particularly in settings where data is structured, by obtaining fewer measurements, demonstrating that low-throughput technologies can be intelligently deployed for large problems. Both strategies have been effective in theory and practice.

Recently, Haupt et. al. [11] and Davenport et. al. [6] showed that *adaptive* data acquisition procedures, which use past measurements to inform subsequent ones, bring performance improvements over the best passive methods. Both results show that, for fixed sensing budget, one can recover a sparse vector much more accurately by adaptively focusing measurements onto components with signal. More broadly, a large body of work has shown that active learning often enjoys significantly better sample complexity bounds than passive methods [10]. While active learning is often quite *statistically* efficient, its role in leading to *computational* gains is largely unexplored.

Motivated by the success of adaptive sensing in sparse vector recovery, we study the low rank matrix completion problem, a natural generalization. This problem involves recovering a low rank matrix from a subset of its entries and has been applied to recommender systems [19], biological sciences [17], network tomography [7], and several other domains. The numerous applications have spurred theoretical study of the problem, and it is now well-known that one can recover a low rank matrix using a vanishingly small fraction of randomly sampled entries [18].

Further generalizing, we also study low rank tensor completion. This problem is more relevant to biomedical imaging and video processing [15] but is far less known than matrix completion. To our knowledge, there are no known statistical guarantees for exact tensor completion.

1.1 Contributions

In this paper, we develop sequential algorithms with adaptive sampling schemes for matrix and tensor completion. The algorithms process the matrix (tensor) in one pass in a streaming fashion, using the subspace detection test of Balzano et. al. [2] to select a few columns to measure in their entirety. The streaming nature of the algorithms lead to substantial benefits in both running time and memory overhead.

In terms of sample complexity, we prove that our algorithm for matrix completion (Algorithm 1) improves on the sample complexity bound of Recht [18], demonstrating the best known upper bound for matrix completion. Adaptive sampling is a instrumental in obtaining our bounds as they are lower than the necessary conditions for matrix completion from randomly sampled entries [5]. This establishes that our procedure outperforms *all* passive matrix completion algorithms.

For the tensor completion problem, we analyze a non-trivial generalization of our matrix completion algorithm and establish sample complexity bounds that scales with the sum of the tensor dimensions. This improves on the most natural generalization of Algorithm 1, which has sample complexity depending on the product of the tensor dimensions. We complement this upper bound with a lower bound for tensor completion under random sampling, showing that our adaptive strategy out-performs *any* passive algorithm. We show that this algorithm is also computationally fairly efficient. These are the first sample complexity upper and lower bounds for exact tensor completion.

Broadly speaking, our work, along with [11, 6] demonstrates that adaptivity can further reduce data acquisition costs, even over the best passive schemes. At the same time, adaptive sampling can lead to computationally efficient algorithms that scale to large data sets. We hope this motivates subsequent study of adaptive data acquisition schemes, both in theory and practice.

2 Related Work

The matrix completion problem has received considerable attention in recent years. A series of papers [4, 9, 14, 5, 18], culminating in Recht's elegent analysis of the nuclear norm minimization program, address the exact matrix completion problem through the framework of convex optimization, establishing that $O(r \max\{\mu_0, \mu_1^2\}(n_1 + n_2) \log^2(n_2))$ randomly drawn samples are sufficient to exactly identify an $n_1 \times n_2$ matrix with rank r. Here μ_0 and μ_1 are parameters characterizing the *incoherence* of the matrix, which we will define shortly. Candes and Tao [5] proved that under random sampling $\Omega(n_1 \mu_0 r \log(n_2))$ samples are necessary, showing that nuclear norm minimization is near-optimal.

Tensor completion is less well known, despite being a natural generalization of matrix completion. The main challenge stems from the NP-hardness of computing most tensor decompositions, pushing researchers to study alternative structure-inducing norms in lieu of the nuclear norm [8, 20]. Both papers derive algorithms for tensor completion, but neither provide sample complexity bounds for the noiseless case, which we study here. Tomioka et. al. [21] do derive bounds for the noisy setting, but to our knowledge, our results are the first such guarantees for exact tensor recovery.

Our approch to the matrix completion problem involves adaptive data acquisition, and consequently our work is closely related to a number of papers focusing on estimating or localizing a sparse, possibly structured, vector corrupted by noise using adaptive measurement design [1, 6, 11, 12]. In these problems, specifically, problems where the sparsity basis is known a priori, we have a reasonable understanding of how adaptive sampling can lead to performance improvements. As a low rank matrix is sparse in its unknown eigenbasis, the completion problem is coupled with learning this basis, which poses a new challenge for adaptive sampling procedures.

Our algorithms, which learn the column space of the matrix by sequentially processing the columns, are also closely related to ideas employed for subspace detection – testing whether a vector lies in a known subspace – and subspace tracking – learning a time-evolving low dimensional subspace from vectors lying close to that subspace. Balzano et. al. [2] prove guarantees for subspace detection with known subspace and a partially observed vector, and we will leverage their results heavily in our analysis. Subspace tracking from partial information has been studied by both He et. al. [13] and Mateos and Giannakis [16], who propose stochastic gradient-style algorithms, but little is known theoretically about this problem.

3 Definitions and Preliminaries

Before presenting our algorithms and theoretical guarantees, we clarify some notation and definitions. Let $M \triangleq U \Sigma V^\star \in \mathbb{C}^{n_1 \times n_2}$ be a rank r matrix of complex entries. We refer to the columns of M as $c_1, \ldots c_{n_2}$.

Let $\mathbb{M} \in \mathbb{C}^{n_1 \times \ldots \times n_T}$ denote an order T tensor of complex entries. The CANDECOMP-PARAFAC decomposition factors $\mathbb{M}$ into a sum of rank one tensors:

$$\mathbb{M} = \sum_{k=1}^{r} a_k^{(1)} \circ a_k^{(2)} \circ \ldots \circ a_k^{(T)} \tag{1}$$

where $\circ$ denotes the outer product. Our results focus on low rank tensors, where rank$(\mathbb{M})$ is the smallest value of r that establishes this equality. Note that the vectors $\{a_k^{(t)}\}_{k=1}^r$ need not be orthogonal, nor even linearly independent $\in \mathbb{C}^{n_t}$, for fixed t.

The mode-t subtensors of $\mathbb{M}$, denoted $\mathbb{M}_i^{(t)}$, $i \in [n_t]$, are order $T-1$ tensors obtained by fixing the ith coordinate of the t-th mode. For example, if $\mathbb{M}$ is an order three tensor, then $\mathbb{M}_i^{(3)}$ are the frontal slices.

We represent a d-dimensional subspace $U \subset \mathbb{C}^n$ as a set of orthonormal basis vectors $U = \{u_i\}_{i=1}^d$ and in some cases as $n \times d$ matrix whose columns are the basis vectors. It will be clear from context which interpretation we are using. Define the **orthogonal projection** onto U as:

$$\mathcal{P}_U v \triangleq \sum_{i=1}^{d} u_i \frac{\langle u_i, v \rangle}{\langle u_i, u_i \rangle^2}$$

Where $\langle \cdot, \cdot \rangle$ is the usual notion of inner product.

For a set $\Omega \subset [n]$, $c_\Omega \in \mathbb{C}^{|\Omega|}$ is the vector whose elements are $c_i, i \in \Omega$ indexed lexicographically. Similarly the matrix $U_\Omega \in \mathbb{C}^{|\Omega| \times d}$ has rows indexed by Ω lexicographically. Note that U_Ω is a $|\Omega| \times d$ matrix with columns $u_{i\Omega}$ where $u_i \in U$, rather than a set of orthonormal basis vectors.

These definitions extend to the tensor setting with slight modifications. We use the `vec` operation to unfold a tensor into a single vector with `refold` as the inverse operation. One readily verifies that $\langle x, y \rangle = \texttt{vec}(x)^T \texttt{vec}(y)$. For a subspace $U \subset \mathbb{C}^{\bigcirc n_i}$, we write it as a $(\prod n_i) \times d$ matrix whose columns are $\texttt{vec}(u_i)$, $u_i \in U$. We can then define projections and subsampling as we did in the vector case.

As in recent work on matrix completion [5, 18], we require a certain amount of incoherence between the subspaces associated with M ($\mathbb{M}$) and the standard basis. We employ the usual definition of **coherence**:

Definition 3.1. *The **coherence** of an r-dimensional subspace $U \subset \mathbb{C}^n$ is:*

$$\mu(U) \triangleq \frac{n}{r} \max_{1 \le j \le n} ||\mathcal{P}_U e_j||^2 \tag{2}$$

where e_j denotes the jth standard basis element.

In previous analyses of matrix completion, the *incoherence assumption* is that both the row space and column space of the unknown matrix have coherences upper bounded by μ_0. Candés and Recht [4] demonstrated that some notion of incoherence is necessary to obtain non-trivial guarantees for matrix completion under random sampling. Even under active sampling, it is impossible to identify a rank one matrix that is zero in all but one entry without observing the entire matrix. Requiring that the matrix subspaces have low coherence precludes these degenerate cases.

Some fairly simple calculations, which we defer to the appendix, reveal some properties of incoherent subspaces, that will play a role in our analysis.

Lemma 3.1. *Let $U_1 \subset \mathbb{C}^{n_1}, U_2 \subset \mathbb{C}^{n_2}, \ldots U_T \subset \mathbb{C}^{n_T}$ be subspaces of dimension at most d, let $W_1 \subset U_1$ have dimension d'. Define $\mathbb{S} = \text{span}(\{\bigcirc_{t=1}^T u_i^{(t)}\}_{i=1}^d)$. Then:*

(a) $\mu(W_1) \le \frac{\dim(U_1)}{d'} \mu(U_1)$.

(b) $\mu(\mathbb{S}) \le d^{T-1} \prod_{i=1}^T \mu(U_i)$.

4 Matrix Completion

Our algorithm for matrix completion sequentially builds the column space of the unknown matrix M by selecting a few columns of the matrix to observe in their entirety. We maintain a candidate column space $\tilde{U}$ and for each column c_i, we test whether $c_i \in \tilde{U}$ or not, choosing to completely observe c_i if it does not lie in $\tilde{U}$. A key insight, which was observed by Balzano et. al. [2], is that this test can be performed reliably

Algorithm 1 Sequential Matrix Completion (M, m)

1. Let $\tilde{U} = \emptyset$.
2. Randomly select $\Omega \subset [n_1]$ with $|\Omega| = m$.
3. For each column c_i of M:
   (a) If $||(I - P_{\tilde{U}_\Omega})c_{i\Omega}||_2^2 > 0$:
      i. $\hat{c}_i \leftarrow c_i$, $u_i \leftarrow \frac{\mathcal{P}_{\tilde{U}^\perp} c_i}{||\mathcal{P}_{\tilde{U}^\perp} c_i||}$.
      ii. $\tilde{U} \leftarrow \tilde{U} \cup \{u_i\}$.
   (b) Otherwise $\hat{c}_i = \tilde{U}(\tilde{U}_\Omega^\star \tilde{U}_\Omega)^{-1} \tilde{U}_\Omega'^\star c_{i\Omega}$
4. Return $\hat{M}$ with $\hat{M}_i = \hat{c}_i$.

by subsampling the coordinates of c_i. The other key insight, is that active sampling can allow us to exactly identify a direction outside of $\tilde{U}$ that belongs to the column space of M.

The formal description of the sequential matrix completion algorithm is given in Algorithm 1. In more detail, at every iteration, we randomly sample a set of m coordinates of the column c_i and compute the projection of $c_{i\Omega}$ onto the orthogonal complement of $\tilde{U}_\Omega$. This projection serves to test whether $c_i \in \tilde{U}$ or not. If it is, then we already have enough information to recover it; if not, we observe the entire column and add it to our subspace $\tilde{U}$.

Computationally, Algorithm 1 is efficient in both time and space. In terms of running time, each iteration of the algorithm predominately operates with matrices of size $|\Omega| \times r$ and vectors of length Ω. The few iterations where we add a column to $\tilde{U}$, we must perform the Gram-Schmidt process, which is also efficient. In contrast, existing algorithms for matrix completion are iterative in nature and compute a singular value decomposition on each iteration, leading to considerable computational overhead [3].

The memory requirements of the algorithm are also minimal; indeed it is sublinear in $n_1 \times n_2$, the number of entries of the matrix. The algorithm streams the columns into memory, maintains the column space $\tilde{U}$ and only needs to maintain the coefficients of each column. This amounts to at most $(n_1 + n_2) \times r$ parameters, implying that the algorithm can scale to incredibly large datasets, as we show in Section 6.

Our main result for matrix completion characterizes the performance of Algorithm 1 in terms of both sample complexity and running time. The only assumption is that the column space of M is *incoherent*.

Theorem 4.1. *Let $M := U\Sigma V^\star \in \mathbb{C}^{n_1 \times n_2}$ have rank r, and fix $\delta > 0$. Assume $\mu(U) \leq \mu_0$. Setting $m \geq 24r^2\mu_0^2 \log(\frac{2r}{\delta})$, Algorithm 1 exactly recovers M with probability at least $1 - 4r\delta + \delta$ while using at most:*

$$24n_2r^2\mu_0^2 \log(2r/\delta) + rn_1 \tag{3}$$

observations. Algorithm 1 runs in $O(n_1n_2r + r^3m)$ time.

We make a few comments about the theorem, before providing a proof sketch. Recht [18] recently guaranteed exact recovery for the nuclear norm minimization procedure as long as the number of observations exceeds $32r(n_1 + n_2)\max\{\mu_0, \mu_1^2\}\beta \log^2(2n_2)$ where β controls the probability of failure and can be thought of as a constant and $||UV^\star||_\infty \leq \mu_1\sqrt{r/(n_1n_2)}$ with μ_1 as another coherence parameter. Without any additional assumptions, μ_1 can be as large as $r\sqrt{\mu_0}$. In this case, our bound matches his exactly except in logarithmic terms and constants, where our bound brings about substantial improvement.

It is also worth noting that we make no assumptions other than the incoherence of the column space U. In particular, the row space V can be highly coherent without affecting the performance of our algorithm. All previous results require that V is also incoherent.

We can also compare our results to known lower bounds. In the passive setting, under uniform sampling, Candes and Tao [5] showed that $\Omega(\mu_0 r n_2 \log(n_1/\delta))$ samples are necessary for completion of a rank r matrix whose subspaces have incoherence bounded by μ_0. Our algorithm improves on this lower bound in the logarithmic factors, demonstrating that active sampling provides advantages over *any* passive algorithm. Unfortunately, just like previous results, Algorithm 1 deviates from the lower bound in its dependence on r and μ_0, and it is an interesting open question to tighten this gap.

In the active setting, a parameter counting argument reveals that a rank r matrix has $r(n_1 + n_2 - r)$ degrees of freedom. Observing the (i, j)th entry M_{ij} of the matrix leads to a polynomial equation of the form $\sum_k u_k(i)\sigma_k v_k(j) = M_{ij}$. If $m < r(n_1 + n_2 - r)$ this system is underdetermined, has M as one solution, so it must have infinitely many solutions, showing that $\Omega(r(n_1 + n_2))$ observations are necessary for exact recovery, even under adaptive sampling. Thus, our algorithm enjoys sample complexity with optimal dependence on matrix dimensions.

It seems that the additional dependence on r in our bound is not an artifact of our analysis, but rather unavoidable for our algorithm and possibly any algorithm that process columns sequentially. This factor arises because with no assumptions, Lemma 3.1(a) is tight, meaning that any individual column can have incoherence $r\mu_0$, even though the column space has incoherence μ_0. If each column was itself incoherent – a much stronger requirement – then our algorithm enjoys an improved sample complexity guarantee.

Proof sketch of Theorem 4.1. The proof of correctness has two components: (1) an analysis of the test, $||(I - \mathcal{P}_{\tilde{U}_\Omega})c_{i\Omega}||^2$, and (2) verification that the remaining columns are recovered exactly. For the former, we verify that, with high probability, the test identifies columns with energy orthogonal to the subspace $\tilde{U}$. Balzano et. al. [2] analyzed this test statistic and we adapt Theorem 1 in that paper to the following lemma, characterizing the behavior of projections under random sampling:

Lemma 4.2. *Suppose that $\tilde{U} \subset U$ and a column $c_j \in U$ but $c_j \notin \tilde{U}$. Let $\delta < 1/e$. If $m \geq 24r^2\mu_0^2 \log(2r/\delta)$ then with probability $\geq 1-4\delta$, $||(I-P_{\tilde{U}_\Omega})c_{j\Omega}||^2 > 0$. If $c_j \in \tilde{U}$ then with probability 1, $||(I-P_{\tilde{U}_\Omega})c_{j\Omega}||^2 = 0$.*

The lemma, which we prove in the appendix, shows that the test statistic can identify the columns with additional information about U using few observations. Since Algorithm 1 fully samples these columns, a union bound verifies that at the end of the algorithm, $\tilde{U} = U$ with high probability.

For step (2), if $c_i \in \tilde{U}$, then as long as $|\Omega|$ is large enough, the matrix $\tilde{U}_\Omega^\star \tilde{U}_\Omega$ will be invertible, so c_i will be recovered exactly. The probability that $\tilde{U}_\Omega^\star \tilde{U}_\Omega$ is invertible is actually subsumed in the probability of success in Lemma 4.2, as invertibility is necessary to compute the projection.

In total, we sample m observations from each column, and completely sample an additional r columns. Setting $m = 24r^2\mu_0^2 \log(2r/\delta)$, gives us the sample complexity bound.

Computationally, the algorithm has three main parts: r inversions of the matrix $U_\Omega^T U_\Omega$ (taking $O(mr^3)$ time), n_2 projection computations (taking $O(n_1 n_2 r)$), and orthonormalizing the basis via the Gram-Schmidt process (taking $O(n_1 r^2)$). □

Algorithm 2 Sequential Tensor Completion $(\mathbb{M}, m_T)$

1. Let $\mathcal{U} = \emptyset$.
2. Randomly select $\Omega \subset \prod_{t=1}^{T-1}[n_t]$ with $|\Omega| = m_T$.
3. For each mode-T subtensor $\mathbb{M}_i^{(T)}$ of $\mathbb{M}$, $i \in [n_T]$:
   (a) If $||\mathbb{M}_{i\Omega}^{(T)} - \mathcal{P}_{\mathcal{U}_\Omega}\mathbb{M}_{i\Omega}^{(t)}||_2^2 > 0$:
      i. $\hat{\mathbb{M}}_i^{(T)} \leftarrow$ recurse on $(\mathbb{M}_i^{(T)}, m_{T-1})$
      ii. $\mathbb{U}_i \leftarrow \frac{\mathcal{P}_{\mathcal{U}^\perp}\hat{\mathbb{M}}_i^{(T)}}{||\mathcal{P}_{\mathcal{U}^\perp}\hat{\mathbb{M}}_i^{(T)}||}$.
      iii. $\mathcal{U} \leftarrow \mathcal{U} \cup \mathbb{U}_i$.
   (b) Otherwise $\hat{\mathbb{M}}_i^{(T)} \leftarrow \mathcal{U}(\mathcal{U}_\Omega^\star \mathcal{U}_\Omega)^{-1}\mathcal{U}_\Omega \mathbb{M}_{i\Omega}^{(T)}$
4. Return $\hat{\mathbb{M}}$ with mode-T subtensors $\hat{\mathbb{M}}_i^{(T)}$.

5 Tensor Completion

With very few adjustments, Algorithm 1 can also be applied to the tensor completion problem. Let $\mathbb{S}$ denote span$(\{\mathbb{M}_i^{(T)}\}_{i=1}^{n_T})$ (the span of the mode-T subtensors). It is easy to see that $\mathbb{S}$ has dimension at most r, since it is also the span of $\{a_k^{(1)} \circ a_k^{(2)} \circ \ldots \circ a_k^{(T-1)}\}_{k=1}^r$. We could build this subspace sequentially, using the same test statistic as before and completely observing each mode-T subtensor that did not completely lie in our current subspace.

Unfortunately, applying the analysis of Theorem 4.1 reveals that the sample complexity of this algorithm is $r(\prod_{t=1}^{T-1} n_t) + n_T r^{2(T-1)}\mu_0^{2(T-1)}\log(2r/\delta)$, under the assumptions that the subspaces $A^{(t)} =$ span$(\{a_i^{(t)}\}_{i=1}^r)$ all have coherence bounded by μ_0. In contrast, a parameter counting argument reveals that there are only $r\sum_{t=1}^T n_t$ degrees of freedom in the tensor, urging us to strive for better algorithms. In particular, we would like to significantly improve our dependence on n_i.

The weakness of the above algorithm is that it does not exploit additional structure in the mode-T subtensors, observing them as needed rather than attempting to complete them. The recursive version of that algorithm, which only fully observes individual fibers of the tensor and completes everything else, has significantly improved sample complexity, and is the algorithm we study in the remainder of this section. The pseudocode is provided in Algorithm 2.

Theorem 5.1. *Let $\mathbb{M} = \sum_{i=1}^r \bigcirc_{t=1}^T a_j^{(t)}$ be a rank r order-T tensor with subspaces $A^{(t)} =$ span$(\{a_j^{(t)}\}_{j=1}^r)$. Suppose that all of $A^{(1)}, \ldots A^{(T-1)}$ have coherence bounded above by μ_0. Set $m_t = 24r^{2(t-1)}\mu_0^{2(t-1)}\log(2r/\delta)$ for each t. Then with probability $\geq 1 - 5\delta T r^T$, Algorithm 2 exactly recovers $\mathbb{M}$ using no more than*

$$24(\sum_{t=1}^T n_t)r^{2(T-1)}\mu_0^{2(T-1)}\log(2r/\delta) \tag{4}$$

observations. The running time of Algorithm 2 is:

$$\tilde{O}\left(r\left(\prod_{i=1}^T n_i\right) + r^{2T+1}\right) \tag{5}$$

Setting $\delta = o(1/r^T)$ we see that with probability $1 - o(1)$ Algorithm 2 recovers $\mathbb{M}$ and uses

$$O(Tr^{2(T-1)}\mu_0^{2(T-1)}(\sum_{t=1}^{T} n_t)\log(r))$$

measurements, which is linear in the dimensions of the tensor, demonstrating a significant improvement over the straightforward application of Algorithm 1 to the tensor case. The dependence on r and μ_0 seem undesirable, but as we will show, these dependencies are almost unavoidable.

Before presenting our lower bound, we briefly compare Theorem 5.1 to existing results for tensor completion. While the problem has not received much attention, there are some algorithms and some theoretical results that we can compare to. The most obvious idea is to unfold the matrix into a $n_1 \times \prod_{t=2}^{T} n_t$ matrix, which must have rank r, and apply any matrix completion algorithm. However, existing lower bounds show that the sample complexity of this approach will scale with $\prod_{t=2}^{T} n_t$, which is much worse than our guarantee.

To our knowledge, the only known theoretical analysis of tensor completion is by Tomioka et. al. [21], who study the noisy version of the problem. They study an optimization involving the nuclear norms of the T different unfoldings of the tensor, but their results imply consistency in Frobenius norm only $\Omega\left(r\prod_{t=2}^{T} n_t\right)$ samples. Again this sample complexity has significantly worse dependence on the tensor dimensions than our algorithm.

Theorem 5.2 (Passive Lower Bound)**.** *Fix $1 \leq m, r \leq \min_t n_t$ and $\mu_0 > 1$. Fix $0 < \delta < 1/2$ and suppose that we do not have the condition:*

$$-\log\left(1 - \frac{m}{\prod_{i=1}^{T} n_i}\right) \geq \frac{\mu_0^{T-1} r^{T-1}}{\prod_{i=2}^{T} n_i} \log\left(\frac{n_1}{2\delta}\right) \tag{6}$$

Then there exist infinitely many pairs of distinct $n_1 \times \ldots \times n_T$ order-T tensors $\mathbb{M} \neq \mathbb{M}'$ of rank r with coherence parameter $\leq \mu_0$ such that $\mathcal{P}_\Omega(\mathbb{M}) = \mathcal{P}_\Omega(\mathbb{M}')$ with probability at least δ. Each entry is observed independently with probability $T = \frac{m}{\prod_{i=1}^{T} n_i}$.

Theorem 5.2 implies that as long as the right hand side of Equation 6 is at most $\epsilon < 1$, and:

$$m \leq n_1 \mu_0^{T-1} r^{T-1} \log\left(\frac{n_1}{2\delta}\right)(1 - \epsilon/2) \tag{7}$$

then with probability at least δ there are infinitely many matrices that agree on the observed entries. This gives a necessary condition on the number of samples required for tensor completion.

We make two remarks about the lower bound:

1. The lower bound holds under the *Bernoulli* sampling model rather than the *Uniform-at-random* model. Candes and Tao [5] show how to translate results for the bernoulli model to the uniform model and since their proof works here as stated, we do not dive into these details.
2. Equation 7 shows that the factors of $r^{T-1}\mu_0^{T-1}$ are necessary, while our guarantee for Algorithm 2 has factors of $r^{2(T-1)}\mu_0^{2(T-1)}$. As before, the extra factors of r arise from the fact that while each subspace is incoherent, each individual sub-tensor may not be; making the stronger assumption that each subtensor is also incoherent would improve the dependence on r.

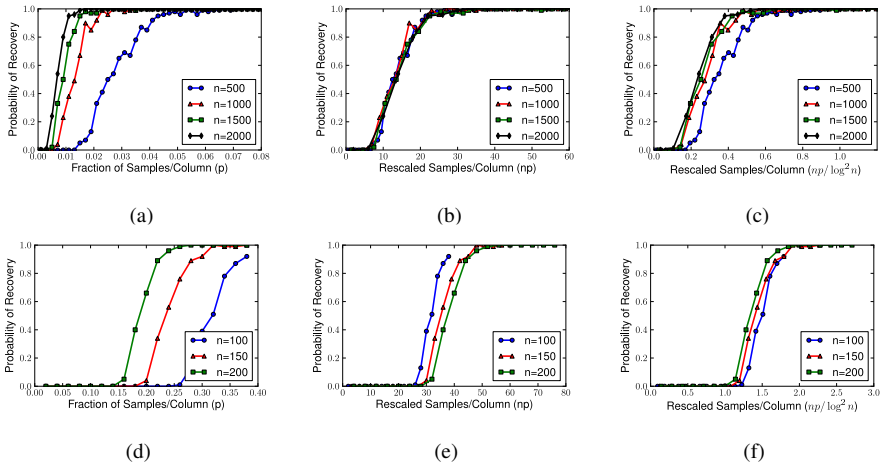

Figure 1: Probability of success curves for Algorithm 1 and SVT. Success probability as a function of: Left: p, the fraction of samples per column, Center: np, total samples per column, and Right: $np \log^2 n$, expected samples per column for passive matrix completion.

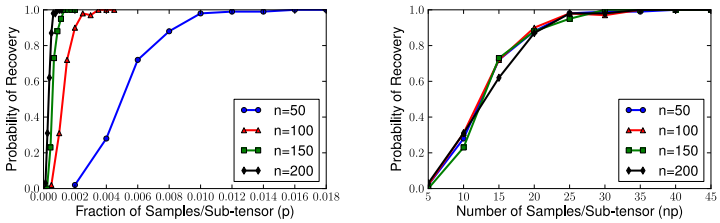

Figure 2: Probability of success curves for Algorithm 2 on order-3 tensors. Success probability as a function of: (a): p, the fraction of samples per subtensor and (b): np, total samples per subtensor.

6 Simulations

We verify some of our theoretical results through several simulations. Our primary goal is to empirically demonstrate the sample complexity guarantees for Algorithm 1 and 2. We also study the scalability of our algorithms and compare to some existing methods for matrix completion.

We verify Algorithm 1's linear dependence on n in Figure 1, where we empirically compute the success probability of the algorithm for varying values of n and $p = m/n$, the fraction of entries observed per column. Here we study square matrices of fixed rank $r = 5$ with $\mu(U) = 1$. Figure 1(a) shows that Algorithm 1 can succeed with sampling a smaller and smaller fraction of entries as n increases, as we expect from Theorem 4.1. In Figure 1(b), we instead plot success probability against total number of observations per column. The fact that the curves coincide suggests that the samples per column, m, is constant with respect to n, which is precisely what Theorem 4.1 implies. Finally, in Figure 1(c), we rescale instead by $n/\log^2 n$, which corresponds to the passive sample complexity bound [18]. Empirically, the fact that these curves do not line up demonstrates that Algorithm 1 requires fewer than $\log^2 n$ samples per column, outperforming the passive bound.

Unknown M				Computational Results	
n	r	m/d_r	m/n^2	time (s)	$\|\|\hat{M} - M\|\|_F$
1000	10	3.4	0.07	16	$2.3e-12$
	50	3.3	0.33	29	$4.3e-12$
	100	3.2	0.61	45	$4.5e-12$
5000	10	3.4	0.01	3	$1.2e-11$
	50	3.5	0.07	27	$3.7e-11$
	100	3.4	0.14	104	$4.4e-11$
10000	10	3.4	0.01	10	$3.6e-11$
	50	3.5	0.03	84	$5.6e-11$
	100	3.5	0.07	283	$6e-11$

Table 1: Computational performance of Algorithm 1 on large low-rank matrices. $d_r = r(2n - r)$ is the degrees of freedom, so m/d_r is the oversampling ratio. m/n^2 is fraction of entries.

The second row of Figure 1 plots the same probability of success curves for the Singular Value Thresholding (SVT) algorithm [3]. As is apparent from the plots, SVT does not enjoy a linear dependence on n; indeed Figure 1(f) confirms the logarithmic dependency that we expect for passive matrix completion. Comparing SVT to Algorithm 1 via these thresholds establishes that Algorithm 1 has empirically better performance.

To confirm Algorithm 1's improvement in computational complexity over existing methods, we ran Algorithm 1 on large-scale matrices, recording the running time and error in Table 1. To contrast with SVT, we refer the reader to Table 5.1 in [3]. As an example, recovering a 10000×10000 matrix of rank 100 takes close to 2 hours with the SVT, while it takes less than 5 minutes with Algorithm 1.

For Algorithm 2, we also confirm the linear scaling on dimension n with the probability of success curves in Figure 2. Here we fixed $m_t = m$ for all t, $r = 5$ and $T = 3$ and plot the probability of success as a function of p, the fraction of entries observed on order-2 tensors and m the total number of entries observed on each subtensor. As before, the fact that the curves line up in Figure 2(b) shows that Algorithm 2 requires a constant number of observations per subtensor, confirming the linear scaling on tensor dimensions.

7 Conclusions and Open Problems

In this work, we demonstrate how algorithms that are both active and sequential can offer significant improvements in time, space, and measurement overhead over passive algorithms for matrix and tensor completion. These algorithms are incredibly appealing from a practical perspective, motivating further study of sequential active algorithms for machine learning.

One short-coming of our work is that we do not study the noisy versions of matrix and tensor completion. Robustifying our algorithms to noise turns out to be non-trivial; while one can analyze the test $||(I - \mathcal{P}_{\tilde{U}_\Omega})c_\Omega||^2$ when the matrix is corrupted, the challenge involves controlling the deviation between the recovered column space $\tilde{U}$ and the the true column space U. We intend to explore this direction further, in hopes of developing a more practical algorithm.

Several interesting theoretical questions arise from our work:

1. Can we tighten the dependence on rank and incoherence parameter for these problems?
2. Can one generalize the nuclear norm minimization program for matrix completion to the tensor completion setting while providing theoretical guarantees on sample complexity?

3. Are there tighter lower bounds for tensor completion under random sampling that scale with the tensor order?

We hope to pursue these directions in future work.

References

[1] S. Balakrishan, M. Kolar, A. Rinaldo, and A. Singh. Recovering block-structured activations using compressive measurements. *Technical Report, arXiv:1209.3431*, 2012.

[2] L. Balzano, B. Recht, and R. Nowak. High-dimensional matched subspace detection when data are missing. In *Proceedings of the IEEE International Symposium on Information Theory*, June 2010.

[3] J. Cai, E. J. Candés, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 2008.

[4] E. J. Candès and B. Recht. Exact Matrix Completion via Convex Optimization. *Foundations of Computational Mathematics*, 9:717–772, 2009.

[5] E. J. Candès and T. Tao. The Power of Convex Relaxation: Near-Optimal Matrix Completion. *IEEE Transactions on Information Theory*, 56:2053–2080, 2010.

[6] M. A. Davenport and E. Arias-Castro. Compressive binary search. In *Proceedings of the IEEE International Symposium on Information Theory*, July 2012.

[7] B. Eriksson, P. Barford, J. Sommers, and R. Nowak. Inferring unseen components in the internet core. *IEEE Journal on Selected Areas of Communication*, 29:1788–1798, October 2011.

[8] S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2), 2011.

[9] D. Gross. Recovering low-rank matrices from few coefficients in any basis. *CoRR*, abs/0910.1879, 2009.

[10] S. Hanneke. Activized learning: Transforming passive to active with improved label complexity. *Journal of Machine Learning Research*, 2012.

[11] J. Haupt, R. Castro, and R. Nowak. Distilled sensing: Adaptive sampling for sparse detection and estimation. *Technical Report, arXiv:1001.5311*, 2010.

[12] J. D. Haupt, R. G. Baraniuk, R. M. Castro, and R. D. Nowak. Compressive distilled sensing: Sparse recovery using adaptivity in compressive measurements. In *Proc. 43rd Asilomar Conf. on Signals, Systems, and Computers*, 2009.

[13] J. He, L. Balzano, and J.C.S. Lui. Online robust subspace tracking from partial information. *Technical Report, arXiv:1109.3827*, 2011.

[14] R. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, June 2010.

[15] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2012.

[16] G. Mateos and G. B. Giannakis. Sparsity control for robust principal component analysis. In *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*, pages 1925 –1929, November 2010.

[17] O. Milenkovic, W. Dai, and N. S. Prasad. Lowrank matrix completion for inference of proteinprotein interaction networks. In *Proceedings of the International Conference of Numerical Analysis and Applied Mathematics*, 2010.

[18] B. Recht. A Simpler Approach to Matrix Completion. *Journal of Machine Learning Research*, 12:3413–3430, 2011.

[19] N. Srebro. *Learning with Matrix Factorizations*. PhD thesis, MIT, 2004.

[20] R. Tomioka, K. Hayashi, and H. Kashima. Estimation of low-rank tensors via convex optimization. *Technical Report, arXiv:1010.0789*, 2011.

[21] R. Tomioka, T. Suzuki, K. Hayashi, and H. Kashima. Statistical performance of convex tensor decomposition. In *Advances in Neural Information and Processing Systems*, 2011.

A Proof of Lemma 3.1

For the first property, since W_1 is a subspace of U_1, $\mathcal{P}_{W_1} e_j = \mathcal{P}_{W_1}\mathcal{P}_{U_1} e_j$ so $||\mathcal{P}_{W_1} e_j||_2^2 \leq ||\mathcal{P}_{U_1} e_j||_2^2$. The result now follows from the definition of incoherence.

For the second property, we instead compute the incoherence of:

$$\mathbb{S}' = \text{span}\left(\left\{\bigcirc_{t=1}^{T} u^{(t)}\right\}_{u^{(t)} \in U_t \forall t}\right)$$

which clearly contains $\mathbb{S}$. Note that if $\{u_i^{(t)}\}$ is an orthonormal basis for U_t (for each t), then the outer product of all combinations of these vectors is a basis for $\mathbb{S}'$. We now compute:

$$\begin{aligned}
\mu(\mathbb{S}') &= \\
&= \frac{\prod_{i=1}^{T} n_i}{\prod_{t=1}^{T} \dim(U_t)} \max_{k_1 \in [n_1], \ldots, k_T \in [n_T]} ||\mathcal{P}_{\mathbb{S}'}(\circ_{t=1}^{T} e_{k_t})||^2 \\
&= \frac{\prod_{i=1}^{T} n_i}{\prod_{t=1}^{T} \dim(U_t)} \max_{k_1, \ldots, k_T} \sum_{i_1, \ldots, i_T} \langle \circ_{t=1}^{T} u_{i_t}^{(t)}, \circ_{t=1}^{T} e_{k_t} \rangle^2 \\
&= \frac{\prod_{i=1}^{T} n_i}{\prod_{t=1}^{T} \dim(U_t)} \max_{k_1, \ldots, k_T} \sum_{i_1, \ldots, i_T} \prod_{t=1}^{T} (u_{i_t}^{(t)\star} e_{k_t})^2 \\
&= \frac{\prod_{i=1}^{T} n_i}{\prod_{t=1}^{T} \dim(U_t)} \prod_{j=1}^{T} \max_{k_j} \sum_{i=1}^{r} (u_i^{(t)\star} e_{k_j})^2 \\
&\leq \prod_{t=1}^{T} \mu(U_t)
\end{aligned}$$

Now, property (a) establishes that $\mu(\mathbb{S}) \leq \frac{r^T}{r} \mu(\mathbb{S}')$ which is the desired result.

B Proof of Theorem 4.1

The proof of correctness begins by ensuring that for every column c_j, if $c_j \notin \tilde{U}$ then $||(I - \mathcal{P}_{\tilde{U}_\Omega}) c_j \Omega||^2 > 0$ with high probability. This is Lemma 4.2 which we prove here. We first state Theorem 1 from [2], which we then adapt to establish our Lemma:

Theorem B.1. *[2] Let U be a d-dimensional subspace of $\mathbb{C}^n$ and $v \in \mathbb{C}^n$. Fix $\delta > 0$ and let Ω be a randomly sampled index set of size $m \geq \frac{8}{3} d\mu(U) \log\left(\frac{2d}{\delta}\right)$. Then with probability at least $1 - 4\delta$:*

$$\frac{m(1-\alpha) - d\mu(U)\frac{(1+\beta)^2}{(1-\gamma)}}{n} ||v - \mathcal{P}_U v||_2^2 \leq ||v_\Omega - P_{U_\Omega} v_\Omega||_2^2 \tag{8}$$

And

$$||v_\Omega - \mathcal{P}_{U_\Omega} v_\Omega||_2^2 \leq (1+\alpha)\frac{m}{n} ||v - \mathcal{P}_U v||_2^2 \tag{9}$$

Where $\alpha = \sqrt{\frac{2\mu(v)^2}{m} \log(\frac{1}{\delta})}$, $\beta = \sqrt{2\mu(v) \log(\frac{1}{\delta})}$ *and* $\gamma = \sqrt{\frac{8d\mu(U)}{3m} \log(\frac{2d}{\delta})}$.

Proof of Lemma 4.2. Noting that $d\mu(\tilde{U}) \leq r\mu_0$, we can immediately apply Theorem B.1 and are left to verify that the left hand side of Equation 8 is strictly positive. Since $c_i \notin \tilde{U}$ we know that $||(I-\mathcal{P}_{\tilde{U}})c_i||^2 > 0$. Then:

$$\begin{aligned}\alpha &= \sqrt{\frac{2\mu(c_i)^2}{m}\log(1/\delta)} \leq \sqrt{\frac{2\mu_0^2 r^2}{m}\log(1/\delta)} \\ &\leq \sqrt{\frac{2\mu_0^2 r^2}{m}\log(1/\delta)} < 1/2\end{aligned}$$

Here we first used $\mu(c_i) \leq r\mu(U)$ since $c \in \text{span}(U)$. Finally, plugging in the definition of m we arrive at $\alpha < 1/2$. For γ we have by Lemma 3.1(a):

$$\gamma = \sqrt{\frac{8d\mu(\tilde{U})}{3m}\log\left(\frac{2d}{\delta}\right)} \leq \sqrt{\frac{8r\mu_0}{3m}\log\left(\frac{2d}{\delta}\right)} \leq \frac{1}{3}$$

and for $(1+\beta^2)$ (as long as $\delta < 1/e$:

$$\begin{aligned}(1+\beta)^2 &\leq (1 + 2r\mu_0\log(1/\delta) + 2\sqrt{2r\mu_0\log(1/\delta)}) \\ &\leq 6r\mu_0\log(1/\delta)\end{aligned}$$

so that:

$$d\mu(\tilde{U})\frac{(1+\beta)^2}{1-\gamma} \leq \frac{3}{2}r\mu(6r\mu\log(1/\delta)) \leq m/2$$

which completes the proof. □

It is easy to see that if $c_i \in \tilde{U}$ then $||(I - \mathcal{P}_{\tilde{U}_\Omega})c_{i\Omega}||^2 = 0$ deterministically and our algorithm does not further sample these columns. We must verify that these columns can be recovered exactly, and this amounts to checking that $\tilde{U}_\Omega^\star U'_\Omega$ is invertible. Fortunately, this was established as a lemma in [2], and in fact, the failure probability is subsumed by the probability in Theorem B.1:

Lemma B.2. *Let $\delta > 0$ and $m \geq \frac{8}{3}r\mu_0\log(2r/\delta)$, Then:*

$$||(\tilde{U}_\Omega^T\tilde{U}_\Omega)^{-1}||_2 \leq \frac{n}{(1-\gamma)m} \tag{10}$$

with probability $\geq 1-\delta$, provided that $\gamma < 1$. In particular $\tilde{U}_\Omega^T\tilde{U}_\Omega$ is invertible.

Now we argue for correctness: there can be at most r columns for which $||(I - \mathcal{P}_{\tilde{U}_\Omega})c_{\Omega i}||^2 > 0$ since $\text{rank}(M) \leq r$. For each of these columns, from Theorem B.1, we know that with probability $1 - 4\delta$ $||(I - \mathcal{P}_{\tilde{U}_\Omega})c_{\Omega i}||^2 > 0$. By a union bound, with probability $\geq 1 - 4r\delta$ all of these tests succeed, so the subspace $\tilde{U}$ at the end of the algorithm is exactly the column space of M, namely U. All of these columns are recovered exactly, since we completely sample them.

There are at most $r+1$ index sets used throughout the algorithm, as we only update Ω when we add a column to $\tilde{U}$. Except for the last index set, the probability that the corresponding matrices $\tilde{U}_\Omega^\star\tilde{U}_\Omega$ are invertible is subsumed by the success probability of Theorem B.1. In other words, the success of the projection test depends on the invertibility of these matrices, so the fact that we recovered the column space U implies that these matrices were invertible. The last such matrix is invertible except with probability δ from Lemma B.2.

If these matrices are invertible, then since $c_i \in \tilde{U}$, we can write $c_i = \tilde{U}\alpha_i$ and we have:

$$\hat{c}_i = \tilde{U}(\tilde{U}_\Omega^\star \tilde{U}_\Omega)^{-1}\tilde{U}_\Omega^\star \tilde{U}_\Omega \alpha_i = \tilde{U}\alpha_i = c_i$$

So these columns are all recovered exactly. This step only adds a factor of δ to the failure probability, leading to the final term in the failure probability of the theorem.

For the running time, per column, the dominating computational costs involve the projection $\mathcal{P}_{\tilde{U}_\Omega}$ and the reconstruction procedure. The projection involves several matrix multiplications and the inversion of a $r \times r$ matrix, which need not be recomputed on every iteration. Ignoring the matrix inversion, this procedure takes at most $O(n_1 r)$ per column for a total running time of $O(n_1 n_2 r)$. At most r times, we must resample and recompute $(U_\Omega^T U_\Omega)^{-1}$, which takes $O(r^2 m)$, contributing a factor of $O(r^3 m)$ to the total running time. Finally, we run the Gram-Schmidt process once over the course of the algorithm, which takes $O(n_1 r^2)$ time. This last factor is dominated by $n_1 n_2 r$.

C Proof of Theorem 5.1

We first focus on the recovery of the tensor in total, expressing this in terms of failure probabilities of the recursions. Then we apply an inductive argument to bound the failure probability of the entire algorithm. Finally we compute the total number of observations. For now, define τ_T to be the failure probability of recovering a T-order tensor.

By Lemma 3.1, the subspace spanned by the mode-T tensors as incoherence at most $r^{T-2}\mu^{T-1}$ and dimension at most r. From Lemma 4.2, we see that with $m \geq 24 r^{2T-2}\mu_0^{2T-2}\log(2r/\delta)$ the projection test succeeds in identifying informative subtensors (those not in our current basis) with probability $\geq 1 - 4\delta$. With a union bound over these r subtensors, the failure probability becomes $\leq 4r\delta + \delta$, not counting the probability that we fail in recovering these subtensors, which is $r\tau_{T-1}$.

For each mode $T-1$ tensor that we have to recover, the subspace of interest has incoherence at most $r^{T-3}\mu^{T-2}$ and with probability $\geq 1 - 4r\delta$ we correctly identify each informative subtensor as long as $m \geq 24 r^{2T-4}\mu^{2T-4}\log(2r/\delta)$. Again the failure probability is $\leq 4r\delta + \delta + r\tau_{T-2}$.

To compute the total failure probability we proceed inductively. $\tau_1 = 0$ since we completely observe any one-mode tensor (vector). The recurrence relation is:

$$\tau_t = 4r\delta + \delta + r\tau_{t-1} \tag{11}$$

which solves to:

$$\tau_T = \delta + 4r^{T-1}\delta + \sum_{t=1}^{T-2} 5r^t\delta \leq 5\delta T r^T \tag{12}$$

We also compute the sample complexity inductively. Let m_T denote the number of samples needed to complete a T-order tensor. Then $m_1 = n_1$ and:

$$m_t = r m_{t-1} + 24 n_t r^{2t-2}\mu_0^{2t-2}\log(2r/\delta) \tag{13}$$

So that m_T is upper bounded as:

$$\begin{aligned} m_T &= r^{T-1}n_1 + \sum_{t=2}^{T} r^{T-t} 24 n_t r^{2(t-1)}\mu_0^{2(t-1)}\log(2r/\delta) \\ &\leq 24(\sum_{t=1}^{T} n_t) r^{2(T-1)}\mu_0^{2(T-1)}\log(2r/\delta) \end{aligned}$$

The running time is computed in a similar way to the matrix case. Assume that the running time to complete an order t tensor is:

$$O(r(\prod_{i=1}^{T} n_i) + \sum_{i=2}^{t} m_i r^{3+t-i})$$

Note that this is exactly the running time of Algorithm 1, so the base case is satisfied.

Per order $T-1$ subtensor, the projection and reconstructions take $O(r\left(\prod_{t=1}^{T-1} n_t\right))$, which in total contributes a factor of $O(r\left(\prod_{t=1}^{T} n_t\right))$. At most r times, we must complete an order $T-1$ subtensor, and invert the matrix $U_\Omega^T U_\Omega$. These two together take in total:

$$O\left(r\left[r(\prod_{t=1}^{T-1} n_t) + \sum_{t=2}^{T-1} m_t r^{3+T-1-t}\right] + r^3 m_T\right)$$

Finally the cost of the Gram-schmidt process is $r^2 \prod_{t=1}^{T-1} n_t$ which is dominated by the other costs.

In total the running time is:

$$\begin{aligned} & O\left(r\left(\prod_{t=1}^{T} n_t\right) + r^2 \prod_{t=1}^{T-1} n_t + \sum_{t=2}^{T} m_t r^{3+T-t}\right) \\ = \; & O\left(r\left(\prod_{t=1}^{T} n_t\right) + \sum_{t=2}^{T} m_t r^{3+T-t}\right) \end{aligned}$$

since $r \leq n_T$. Now plugging in that $m_i = \tilde{O}(r^{2(i-1)})$, the terms in the second sum are each $\tilde{O}(r^{T+t+1})$ meaning that the sum is $\tilde{O}(r^{2T+1})$. This gives the computational result.

D Proof of Theorem 5.2

We start by giving a proof in the matrix case, which is a slight variation of the proof by Candes and Tao [5]. Then we turn to the tensor case, where only small adjustments are needed to establish the result. We work in the Bernoulli model, noting that Candes' and Tao's arguments demonstrate how to adapt these results to the uniform-at-random sampling model.

D.1 Matrix Case

In the matrix case, suppose that $l_1 = \frac{n_1}{r}$ and $l_2 = \frac{n_2}{\mu_0 r}$ are both integers. Define the following blocks $R_1, \ldots R_r \subset [n_1]$ and $C_1, \ldots C_r \subset [n_2]$ as:

$$\begin{aligned} R_i &= \{l_1(i-1)+1, l_1(i-1)+2, \ldots l_1 i\} \\ C_i &= \{l_2(i-1)+1, l_2(i-1)+2, \ldots l_2 i\} \end{aligned}$$

Now consider the $n_1 \times n_2$ family of matrices defined by:

$$\mathcal{M} = \{\sum_{k=1}^{r} u_k v_k^\star | u_k = [1, \sqrt{\mu_0}]^n \circ \mathbf{1}_{R_k}, v_k = \mathbf{1}_{C_k}\} \tag{14}$$

$\mathcal{M}$ is a family of block-diagonal matrices where the blocks have size $l_1 \times l_2$. Each block has constant rows whose entries may take arbitrary values in $[1, \sqrt{\mu_0}]$. For any $M \in \mathcal{M}$, the incoherence of the column space can be computed as:

$$\begin{aligned}\mu(U) &= \frac{n_1}{r} \max_{j \in [n_1]} ||\mathcal{P}_U e_j||_2^2 = \frac{n_1}{r} \max_{k \in [r]} \max_{j \in [n_1]} \frac{(u_k^T e_j)^2}{(u_k^T u_k)^2} \\ &\leq \frac{n_1}{r} \max_{k \in [r]} \frac{\mu_0}{(n_1/r)} = \mu_0\end{aligned}$$

A similar straightforward calculation reveals that the row space is also incoherent with parameter μ_0.

Unique identification of M is not possible unless we observe at least one entry from each row of each diagonal block. If we did not observe an entry in one such row, then we could vary that corresponding coordinate in the appropriate u_k and find infinitely many matrices $M' \in \mathcal{M}$ that agree with our observations, have rank and incoherence at most r and μ_0 respectively. Thus, the probability of successful recovery is no larger than the probability of observing one entry of each row of each diagonal block.

The probability that any single row of any block is unsampled is $\pi_1 = (1-p)^{l_2}$ and the probability that all of the rows are sampled is $(1-\pi_1)^{n_1}$. This quantity must upper bound the success probability $1-\delta$. Thus:

$$-n_1 \pi_1 \geq n_1 \log(1-\pi_1) \geq \log(1-\delta) \geq -2\delta$$

or $\pi_1 \leq 2\delta/n_1$ as long as $\delta < 1/2$. Substituting $\pi_1 = (1-p)^{l_2}$ we obtain:

$$\log(1-p) \leq \frac{1}{l_2} \log\left(\frac{2\delta}{n_1}\right) = \frac{\mu_0 r}{n_2} \log\left(\frac{2\delta}{n_1}\right)$$

as a necessary condition for unique identification of M.

Exponentiating both sides, writing $p = \frac{m}{n_1 n_2}$ and the fact that $1 - e^{-x} > x - x^2/2$ gives us:

$$m \geq n_1 \mu_0 r \log\left(\frac{n_1}{2\delta}\right)(1 - \epsilon/2)$$

when $\mu_0 r / n_2 \log(\frac{n_1}{2\delta}) \leq \epsilon < 1$.

D.2 Tensor Case

Fix T, the order of the tensor and suppose that $l_1 = \frac{n_1}{r}$ is an integer. Moreover, suppose that $l_t = \frac{n_t}{\mu_0 r}$ is an integer for $1 < t \leq T$. As in the matrix case, we will define a set of blocks, one for each mode of the tensor and the family of tensors

$$\begin{aligned}B_i^{(t)} &= \{l_t(i-1)+1, l_t(i-1)+2, \ldots, l_t i\} \forall i \in [r], t \in [p] \\ \mathcal{M} &= \left\{ \sum_{i=1}^{r} \bigcirc_{t=1}^{T} a_i^{(t)} \;\middle|\; \begin{array}{l} a_i^{(1)} = [1, \sqrt{\mu_0}]^n \circ \mathbf{1}_{B_i^{(t)}} \\ a_i^{(t)} = \mathbf{1}_{B_i^{(t)}}, 1 < t \leq T \end{array} \right\}\end{aligned}$$

This is a family of block-diagonal tensors and just as before, straightforward calculations reveal that each subspace is incoherent with parameter μ_0. Again, unique identification is not possible unless we observe at least one entry from each row of each diagonal block. The difference is that in the tensor case, there

are $\prod_{i \neq 1} l_i$ entries per row of each diagonal block so the probability that any single row is unsampled is $\pi_1 = (1-p)^{\prod_{i \neq 1} l_i}$. Again there are n_1 rows and any algorithm that succeeds with probability $1-\delta$ must satisfy:

$$-n_1 \pi_1 \geq n_1 \log(1-\pi_1) \geq \log(1-\delta) \geq -2\delta$$

Which implies $\pi_1 \leq 2\delta / n_1$ (assuming $\delta < 1/2$). When we substitute in the definition of π_1 we have:

$$\log(1-p) \leq \frac{1}{\prod_{i \neq j} l_i} \log\left(\frac{2\delta}{n_1}\right) = \frac{\mu_0^{T-1} r^{T-1}}{\prod_{i \neq j} n_i} \log\left(\frac{2\delta}{n_1}\right)$$

The same approximations as before yield the bound:

$$m \geq n_1 \mu_0^{T-1} r^{T-1} \log\left(\frac{n_1}{2\delta}\right)(1-\epsilon/2)$$

as long as $\frac{\mu_0^{T-1} r^{T-1}}{\prod_{i \neq j} n_i} \log(\frac{n_1}{2\delta}) \leq \epsilon < 1$.